Long Read based Human Genomic Structural Variation Detection with cuteSV

Tao Jiang[1], Bo Liu[1], Yue Jiang[2], Yan Gao[1], Zhe Cui[1], Yadong Liu[1] and Yadong Wang[1, *]

[1] Center for Bioinformatics, School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

[2] Nebula Genomics, Harbin, Heilongjiang 150030, China

[*] Correspondence: ydwang@hit.edu.cn

Abstract

Long-read sequencing technologies enable to comprehensively discover structural variations (SVs). However, it is still non-trivial for state-of-the-art approaches to detect SVs with high sensitivity or high performance or both. Herein, we propose cuteSV, a sensitive, fast and lightweight SV detection approach. cuteSV uses tailored methods to comprehensively collect various types of SV signatures, and a clustering-and-refinement method to implement a stepwise SV detection, which enables to achieve high sensitivity without loss of accuracy. Benchmark results demonstrate that cuteSV has better yields on real datasets. Further, its speed and scalability are outstanding and promising to large-scale data analysis. cuteSV is available at https://github.com/tjiangHIT/cuteSV.

Keywords

Structural variants detection, Long-read sequencing, High sensitivity, Memory efficient

Background

Structural variations (SVs) represent genomic rearrangements, such as deletions, insertions, inversions, duplications, and translocations at least 50 bp in size [1]. As the largest divergent base pairs across human genomes [2], SVs play a vital role in the human disease (e.g. inherited diseases [3-5] and cancers [6]), evolution (e.g. gene losses and transposon activity [7, 8]), gene regulation (e.g. rearrangements of transcription factors [9]) and other phenotypes in many species (e.g. mating and intrinsic reproductive isolation [10, 11]).

Many efforts have been made toward SV calling along with the wide application of next-generation sequencing (NGS) technologies [12, 13]. The main-stream NGS based methods discovers SVs through read-depth [14], read-pair [15], split-read [16], assembly [17] and the combination of them [18-20]. With the aid of these relevant algorithms, many genomic researches, such as the 1000 Genome Project [1], have made great improvements in structural variant demography, functional impact and disease association. However, the limitation of read length (a few hundred nucleotides at most) has a strong effect on the SV detection sensitivity [21] and might produce a large amount of false-positive calls [22].

Recently, with the rapid development of third-generation sequencing (TGS) technologies, such as Pacific Bioscience (PacBio) [23] and Oxford Nanopore Technology (ONT) [24], it is accessible to obtain the long-range spanning information via TGS. The average read length of the latest long-read sequencing can be up to 10 kbp, which brings a lot of potential for SVs detection at a higher resolution [25]. However, a challenge remains for TGS that the high error rates of 5-20% [26] might give rise to inaccurate read mapping and even result in no mapping, thus leads to poor performance on SV calling.

To take full advantage of long-range spanning information and overcome the influence of sequencing errors, several long-read mapping based SV callers, such as PB-Honey [27], SMRT-SV [28], Sniffles [29], PBSV (https://github.com/PacificBiosciences/pbsv) , SVIM [30], etc., have been developed. These methods usually demand a high-quality alignment as input to implement SV calling. PB-Honey and SMRT-SV prefer to handle alignments generated from BLASR [31]. Sniffles, PBSV, and SVIM tend to manipulate alignments generated from the latest long-read aligners like NGMLR [29], Minimap2 [32], and PBMM2 (https://github.com/PacificBiosciences/pbmm2) . They either identify genome regions with abnormal erroneous, or analyze the soft-clipped read tails, or cluster the variant-spanning signatures, to build probable SV events as much as possible [27-30].

Nevertheless, current long-read SV calling methods remain drawbacks: (1) some approaches like SMRT-SV and PB-Honey can only analyze one specific type of sequencing data since the restrictions on preset data; (2) some methods like rMETL [33] and npInv [34] just handle a subset or a particular class of SVs due to their specific design; (3) some tools like PBSV and SMRT-SV consume large amounts of computational resources in analysis of plenty SV associated attributes, local assembly, and consensus generation; (4) the sensitivity of detection in most algorithms is still fairly low and require higher coverage data to guarantee the performance acceptably. In some sense, all the above shortcomings might restrict the application of long-read SV callers in routine research and clinical practice.

To address these challenges, we present a sensitive, fast and lightweight computational method----- cuteSV. cuteSV is able to discover deletions, insertions, duplications, inversions, and translocations through a stepwise refinement of comprehensive signatures clustering strategy. And it has excellent speed and scalability to large-scale TGS platform-independent data analysis. The evaluation results on several public genuine human datasets show that cuteSV achieves better SVs detection yields compared to the state-of-the-art SV callers, and it is a high-efficiency approach within less memory consumption and competitive running speed. Also, it would save the cost of sequencing and analysis for a single

sample considering the lower demands on sequencing coverage. We anticipate that the widespread use of cuteSV will promote the vigorous development of genome research.

Results

Overview of cuteSV

CuteSV identifying SVs implements four major steps (a schematic illustration is in Figure 1):

(1) Main-stream TGS-based aligners are employed to perform long-read mapping, and Samtools [35] is implemented to finish the conversion, coordinate sorting and index building of alignments.

(2) cuteSV identifies and collects various types of potential SV signatures comprehensively from spatial features, such as coordinates, orientations, chromosomes, etc., in intra- and inter-alignments.

(3) cuteSV completes a stepwise SV detection through clustering and refinement of SV signatures in each read to identify and screen possible candidates.

(4) cuteSV calculates breakpoints and sizes of predicted SVs, and optionally genotypes them according to the ratio of allele counts.

cuteSV also supports SVs detection for multi-process through long-read alignment block division. For more details please see Materials and methods.

SV detection with the HG002 PacBio data

We compared the performance of cuteSV together with state-of-the-art SV callers, i.e. Sniffles, PBSV, and SVIM, on a 69× HG002 PacBio CLR dataset [36] (mean read length, 7938 bp) in the first place. A high-confidence insertion and deletion callset for this sample from the Genome in a Bottle Consortium (GIAB) [37] was selected as the gold standard. A third-party SV comparison tool, Truvari (https://github.com/spiralgenetics/truvari), was used to assess the precision, recall, and F-measure (or F1) in SVs detection. The benchmarking results are shown in Figure 2A. In terms of the 69× CLR data, cuteSV achieved the best precision, recall, and F1, and surpassed 94% together. Sniffles and PBSV were equally powerful in F1 (i.e. 90.22% and 91.48% for Sniffles and PBSV, respectively), and a little behind cuteSV since their relative low recall (i.e. 86.27% and 88.42% for Sniffles and PBSV, respectively). The performance of SVIM was the most unfavorable one since its negative accuracy (i.e. 75.13%) even though it acquired relatively higher sensitivity (i.e. 89.30%). We also down-sampled the original data to assess the performance in lower coverage cases. cuteSV retained significant advantages in recall and F1 and achieved the best performance at every coverage data. Especially at relatively low coverage (i.e. 20×), cuteSV still accomplished over 90% in precision, recall, and F1, respectively. By contrast, it is still a challenging task for other state-of-the-art tools that they required 20 to 40× sequencing coverage and just accomplished over 80% in the recall.

Apart from PacBio CLR data, we assessed the performance with the new PacBio data type HiFi reads (Single-Molecule High-Fidelity reads) [38] as well (see Figure 2B). For the 28× HiFi data [39] (mean read length: 13478 bp), cuteSV and PBSV achieved almost the same results (i.e. precision, 95.2%; recall, 98.0%; F1, 96.6%). Sniffles and SVIM were a little behind them 1% to 4% for corresponding metrics. In terms of 10× HiFi data, unfortunately, the sensitivity of cuteSV was nearly 3% below PBSV, but it was still 93.79% and surpassed Sniffles and SVIM. It was mainly due to the setting of minimal signature size for processing in cuteSV (30 bp in default) was longer than it in PBSV (20 bp in default) exactly, which disabled cuteSV to identify more SVs using evidence in a shorter size. Nevertheless, with the latest HiFi data, we found that the minimum requirement of sequencing coverage had been down to about

5× for cuteSV and PBSV because both achieved more than 91% in precision, sensitivity, and F1, respectively. Overall, cuteSV and PBSV had close power in SVs detection on HiFi data and got one step ahead of Sniffles and SVIM.

The GIAB evidence callsets help construct a baseline of assessment, which describes the performance on identifying insertions and deletions in the high confidence regions [37]. For other classes of SVs, such as inversions, duplications, and translocations, however, it is hard to assess since the lack of corresponding ground truth. Here, we used the Mendelian-Discordance-Rate (MDR), a description of how much percentage of structural variants presented in offspring that did not make an appearance in either biological parent to assess Mendelian consistency. In this end, we used the well-studied GIAB Ashkenazi Trio [36] (see Figure 2C). cuteSV had the lowest MDRs for almost of all the SV types with both two employed aligners (i.e. DEL: 0.103 and 0.072; INS/DUP: 0.162 and 0.130; INV: 0.061 and 0.071; BND: 0.127 and 0.083; All Types: 0.131 and 0.104, for PBMM2 and NGMLR, respectively). The MDRs of Sniffles were slightly lower than that of cuteSV (i.e. All Types: 0.173 and 0.175 for PBMM2 and NGMLR, respectively), whereas PBSV (i.e. All Types: 0.590 and 0.564 for PBMM2 and NGMLR, respectively) and SVIM (i.e. All Types: 0.240 and 0.395 for PBMM2 and NGMLR, respectively) fell behind visibly. Especially, PBSV and SVIM had huge MDRs in the specific SV type, that was, 0.72 in translocations on PBSV and 0.82 in inversions on SVIM, on average.

Moreover, we evaluated the runtime and memory consumption using the 69× HG002 CLR dataset to show the characteristics of computing power (see Figure 2D and Figure 2E). For runtime in a single thread, Sniffles had the fastest running speed (i.e. 367 minutes) compared to other methods. SVIM (i.e. 423 minutes), cuteSV (i.e. 983 minutes), and PBSV (i.e. 1715 minutes) were in order after Sniffles. When using multi threads, the runtime of cuteSV was on the decrease significantly as the thread increased. Notably, cuteSV only took less than an hour and a half in SVs detection with 16 threads. For memory usage, state-of-the-art methods occupied more than a dozen GB, i.e. 17.61 GB on Sniffles, 10.57 GB on PBSV, and 9.41 GB on SVIM. While cuteSV only used 0.35 GB no matter how many threads in use, that was, cuteSV saved 96% to 98% of memory and was a memory-efficient method.

Comparison of SV calling with different aligners

Long-read mapping is a much essential step in genetic analysis, which brings abundant variant associated evidence for discovering SVs. While using various aligners based on heterogeneous algorithms might result in different predictions. To estimate what influence of long-read aligners in SV calling, we used two state-of-the-art long-read mappers, i.e. PBMM2 and NGMLR, to perform alignment on the 69× HG002 CLR data.

Firstly, we evaluated the effect on SVs detection sensitivity. We collected SV calls marked as false-negative (FN) by Truvari on different aligners and calculated the difference between them (see Figure 3A and Supplementary Table 1). It demonstrated that PBMM2 was more fitted for the task of discovering less than 1000 bp insertions and deletions since the smaller number of FN calls compared to NGMLR (i.e. 134, 200, 404 on cuteSV, Sniffles, and PBSV, respectively). Especially, PBSV missed 436 more FN insertions in size of 200 to 700 bp when using NGMLR, and 84.63% (369 out of 436) of them coincided with Alus (one type of mobile element about 300 bp in length). On the contrary, NGMLR enabled SV callers to create less FN calls in length of over 1000 bp (i.e. 109, 283, 127 on cuteSV, Sniffles, and PBSV, respectively). For SVIM, there was no significant impact on its sensitivity when using different aligners, due to the slight distinction on FN calls (i.e. 21 more FNs on NGMLR and 10 more FNs on PBMM2, respectively).

Next, we assessed the impact on the accuracy through the different number of false-positive (FP) calls (see Figure 3B and Supplementary Table 2). In terms of less than 1000 bp insertions and deletions, PBMM2 raised more FP calls within cuteSV and Sniffles (i.e. 148 on cuteSV and 65 on Sniffles), whereas NGMLR enabled SVIM to generate much more FP insertions (i.e. 10578 totally). As for PBSV, there was no significant distinction between the two employed mappers since only 31 different FPs in quantity. For SVs larger than 1000 bp, the number of FP calls on cuteSV, Sniffles, and PBSV were similar between the two employed aligners (i.e. 2 on cuteSV, 27 on Sniffles, and 13 on PBSV, respectively). However, 939 more FP calls on SVIM with NGMLR were found, which decreased the precision dramatically.

In view of Mendelian consistency, we compared the effect of aligner selection in MDR as well (see Figure 3C). For all approaches, using NGMLR could result in the lowest MDRs in identifying deletions and translocations significantly, which degraded 0.35% to 7.52%. For the combination of insertions and duplications, and inversions, we could not find a significant bias in both aligners. In terms of total calls, NGMLR could help cuteSV and PBSV to generate variants with higher Mendelian consistency due to the 2% to 3% of MDR reduction. However, PBMM2 was a better choice for SVIM since the over 15% MDR reduction. Sniffles was compatible with these two mappers because there was no obvious difference between them.

Benchmarking of Parameter optimization

To make certain the influence of parameter settings and provide advice of implementation of cuteSV in practice, we assessed the performance of variant calling under two pivotal parameters setting, i.e. –min_support (-s) and –min_size (-l), on multiple coverages of HG002 CLR datasets.

The parameter of –min_support indicates the minimal number of reads supporting an SV that to be reported. From the results shown in Figure 4A and Supplementary Table 3, keeping–min_size (with 30 bp as the default) constant, cuteSV accomplished the best performance when set -s1 to -s10 at 5×, 10×, 20×, 30×, 40×, and 69× coverage, respectively. And a clear trade-off between precision and recall on –min_support setting was observed, where set a lower number could enhance a higher sensitivity whereas might decrease precision respectively, and vice versa.

Another parameter of –min_size means the minimal size of SV signature considered in clustering. We compared the performance between two different –min_size (i.e. -l 30 and -l 50) settings under the same –min_support (see Figure 4B). At various coverages, cuteSV succeed 0.38% to 1.71% more in accuracy using –min_size 50, while 0.06% to 2.19% more recall rates were observed in –min_size 30. Hence, one another obvious trade-off between precision and recall was implied as well that a smaller number of –min_size setting improved sensitivity and brought accuracy down, and vice versa. Nevertheless, for F-measure, the scores were close to each other at each depth and only less than 1% of the change rate was discovered.

SV detection with the NA19240 PacBio CLR data

To assess algorithm performance using only one human sample is not enough. To this end, we selected one another well-studied NA19240 human sample in PacBio CLR sequencing technology (mean read length, 6503 bp, coverage, 40x) [40] from the Human Genome Structural Variation Consortium (HGSV). A callset for this sample was also obtained from the study in [41] as ground truth.

We benchmarked the precision, recall, and F1 and summarized the results in Figure 5A and Supplementary Table 4. cuteSV had the highest recall rates (DEL, 62.79%; INS/DUP, 55.68%; INV,

192 19.11%; All Types, 58.64%) and F1 scores (DEL, 63.01%; INS/DUP, 54.28%; INV, 17.36%; All Types,
193 57.66%). PBSV, Sniffles, and SVIM were behind cuteSV, in turn. One point that could not be ignored
194 was that the precisions in cuteSV were below the best one in each tag a bit (3.20% for DEL; 2.19% for
195 INS/DUP; 19.39% for INV; 3.22% for All Types). However, it was excusable, considering the more TP
196 calls in cuteSV than the corresponding best one, which surpassed 208, 768, 49 and 1016 for DEL,
197 INS/DUP, INV, and All Types, respectively.
198 Moreover, each tool achieved more outstanding performance in deletions (about 60% on average) but
199 relatively poor one in inversion, which was less than 20% in F1. It was also noted that the evaluation
200 results on the NA19240 were far below them on the HG002. This was mainly due to the relatively low
201 quality of ground truth and the lack of high confidence regions about this sample for a highly accurate
202 assessment. Nevertheless, cuteSV was still a promising approach since its superior sensitivities and F1
203 scores.

204 **Comparison of PacBio and ONT sequencing data**

205 cuteSV can detect SVs using not only PacBio but also ONT datasets. To investigate the performance of
206 cuteSV on these two different sequencing technologies, we utilized a 55× PacBio CLR dataset (mean
207 read length, 4334 bp [42]) and a 28× Oxford Nanopore dataset (mean read length, 6432 bp [43]) on the
208 NA12878 human sample. cuteSV recognized 23791 SVs from the PacBio CLR data and 36624 SVs from
209 the ONT data (see Supplementary Table 5). We found 6972 or 29.31% SV calls were unique to PacBio
210 data and 19202 or 52.43% SV calls were unique to ONT data. We further analyze the whole unique calls
211 on different SV types and ascertained that 54.29% (3785 out of 6972) of PacBio-only calls were
212 insertions and 76.28% (14647 out of 19202) of ONT-only calls were deletions. As the error models
213 shown in previous studies, raw long reads have high error rates in false insertions for PacBio [23] and in
214 false deletions for ONT [24]. Hence, the results of a large fraction of unique calls were also reasonable.
215 Moreover, a published callsets of NA12878, which containing 2676 deletions and 68 insertions, was
216 selected as the ground truth to estimate the performance of cuteSV (see Figure 5B). For the total of 43295
217 SVs yielded together on different sequencing data, 94.86% calls in ground truth were found by both
218 datasets, and 3.10% calls in ground truth could be detected through only one data as well. Only 2.04%
219 calls in ground truth were unable to identify. We could confirm the finding in [30] that the vast majority
220 (97%) of ground truth could be identified by PacBio or ONT sequencing technologies.

221 **Discussion**

222 Collection and interpretation of evidence from the long-read alignments is a promising strategy for
223 detecting high-quality SVs since it scans the signatures of fragments changing and constructs putative
224 candidate of SVs. However, under the circumstances of complex SV events and high sequencing error
225 rates, this task is non-trivial in practice.
226 (1) For some regions containing multi types of SV, it is inefficient for restoring a specific class of SV
227 without considering the signature category and sometimes might generate erroneous predictions. (2)
228 Similarly, for some regions containing multi SV alleles, it is too hard to figure out the correspondences
229 between the real allele and its supporting evidence, due to a large amount of less varied signatures mixing
230 in a limited genomic region. (3) Moreover, owing to the high sequencing errors, quite a lot of mistakes
231 might be made in long-read mapping. As a result, some variant associated signatures would change, and
232 some false evidence might generate.

All the above bottlenecks could decrease the performance of sensitive SVs detection. To break through these challenges, we propose a sensitive, fast and lightweight computational method, cuteSV, for sensitively and robustly detecting SVs using long-read. The most critical factor of cuteSV to guarantee the high sensitivity is its stepwise-refinement strategy on clustering through comprehensive signatures and presupposed polymorphic alleles model.

(1) Since the sequencing errors or some other variants participating in, a real SV signature, especially that coming from complex SV, might be split into two parts or more. It is a challenging task to extract and reconstruct the continuous and complete variant-spanning fragment without using local assembly. cuteSV integrates the same type of signatures, which are distributed in a local area densely, to recover and enrich the real evidence which might enhance discovering SVs in complex sequence regions of the genome effectively.

(2) In the genome region with multi non-reference alleles, various allele-supporting signatures and errors, which share a similar breakpoint, are interwoven in a regional cluster. Some real SV alleles with less supporting signatures might be discarded due to insufficient evidence, especially for low coverage sequencing data. To this end, cuteSV proposes a presumed polymorphic alleles model for identifying all potential non-reference alleles through setting a specific ratio of size partitioning to re-cluster each signature and purifying the consistent features. This approach can further improve the recognizing sensitivity of SVs with multiple non-reference alleles.

cuteSV also has a prominent computational performance with its block analysis design. One of the great benefits is that the maximum memory usage is under control by setting the amount of data in the process. Another advantage is that cuteSV can manipulate signatures extraction and clustering in parallel efficiently since the independence of sequencing data. The implementation of both two benefits enables cuteSV to perform SV calling more feasibly just on a desktop computer as well.

With the assistance of high confidence callsets released by several previous studies [37, 41], both sensitivity and specificity are assessed objectively on state-of-the-art SV callers and cuteSV. In most of the datasets, cuteSV discovers more SV calls consistent with the ground truth without loss of accuracy and achieves better F-measure scores. Meanwhile, this advantage is significant no matter what the platforms and SV types are. Besides, cuteSV has better Mendelian consistency among offspring and its parents than other methods. These suggest that cuteSV has a promising capability to identify genomic SVs sensitively and efficiently.

Another important feature of cuteSV is that it can identify SVs with high sensitivity and accuracy on low coverage datasets. TGS has been in development for more than a decade, however, the cost is still more expensive compared to NGS. cuteSV can improve SVs detection on low coverage data through refines signatures in each read and self-adapting clustering, which ensures the high specificity and sensitivity. It is certainly a great improvement which might save considerable sequencing expensive.

Although cuteSV is a promising SV caller with high sensitivity, there are still some SV calls in high confidence callsets that cannot be detected (see Supplementary Figure 1 and 2). From these two examples, we find that the false-negative calls are produced mainly due to the lack of corresponding evidence that supports the ground truth. More specifically, there is a huge difference in size or breakpoint of SV signatures from alignments which enables cuteSV to generate absurd SV calls unexpectedly. These FN calls also happen in Sniffles, PBSV, and SVIM, respectively. So, these could be plausible as well even though losing a few FN calls. Moreover, it remains a lot of work for us to promote cuteSV, such as the generation of the consensus sequence of a specific allele and further optimization of the genotype assignment model. We will continue to refine these new functionalities and carry out progressively in

subsequent releases of cuteSV.

Conclusion

Long-read sequencing technologies provide us with an unprecedented view of genetic variation in the human genome, and obtain unsurpassed achievements. Despite the long success, quite a lot of SVs still have not to be identified even for advanced long-read based algorithms. Therefore, there is an urgent need to address more sensitive SVs detection.

In this paper, we provide a sensitive, fast and lightweight approach to discover high-quality SV with long sequencing reads. It employs a stepwise refinement clustering algorithm to process the comprehensive signatures from inter- and intra-alignment, construct and screen all possible alleles thus completes high-quality SV calling. Benchmark results shows that cuteSV achieves superior sensitivity, accuracy, and power compared to existing methods even on low coverage sequencing data. We expect it brings a lot of potentials for cuteSV to improve the vigorous advancement of genome research.

Materials and methods

Long-read mapping

As a sensitive SV caller, cuteSV imputes SV candidates mainly from the signatures implied in intra- and inter-alignments. We investigate start-of-the-art aligners, which contain high-quality base calling (nucleotide-level alignment) and abundant supplementary alignments, to perform long-read mapping. After long-read mapping, Samtools is used to perform the alignment format conversion, coordinate sorting and index building. Then input the coordinate-sorted BAM file into cuteSV to analyze SVs.

Discovering SV signatures

cuteSV complete signatures discovering mainly from two types of alignment, i.e. intra-alignment and inter-alignment. For intra-alignment, if a deletion or insertion that over 30 bp in length was found in CIGAR filed, a corresponding SV signature will be recorded as a triple, such as

$$SIG = (pos, len, ID) \tag{1}$$

where pos means the starting coordinate on the reference genome, len is the size of corresponding SV, and ID implies the unique read ID. Considering the high error rates of sequencing, cuteSV clusters two signatures SIG_1 and SIG_2 into a same group spontaneously when meeting

$$\begin{cases} pos_2 - pos_1 \leq S_{dis} & if\ INS \\ pos_2 - (pos_1 + len_1) \leq S_{dis} & if\ DEL \end{cases} \tag{2}$$

where S_{dis} is a threshold value to control the integration (500 bp, in default). After integration, a modifying signature SIG_M will be generated as

$$SIG_M = (pos_1, len_1 + len_2, ID) \tag{3}$$

cuteSV collects all integrated and remained SV signatures as the part of evidence for inference of deletions and insertions.

For inter-alignment, e.g. a read in primary alignment and with one supplementary alignment at least, we define a hexagram to describe each of segment in split alignments, which is

$$Seg = (read_s, read_e, Ref_s, Ref_e, Chr, Strand) \quad (4)$$

where $read_s$, $read_e$, Ref_s, Ref_e, Chr, $Strand$ indicates the starting and termination coordinate on read and reference genome, the aligned chromosome ID and orientation, respectively. The SV event was just hided in the $Segs$ combination. cuteSV applies various heuristic rules to construct the potential SV signatures as below:

(1) If two adjacent segments on the read level are aligned to the same chromosome and share the same orientation, cuteSV will compare the $Diff_{dis}$, a description of the distance on the genome level minus the distance on the read level, which is

$$Diff_{dis} = (Ref_{2s} - Ref_{1e}) - (read_{2s} - read_{1e}) \quad (5)$$

Besides, cuteSV calculates the $Diff_{olp}$, a description of how much the intersection is of the two adjacent segments, which is

$$Diff_{olp} = Ref_{1e} - Ref_{2s} \quad (6)$$

When $Diff_{olp} < 30\ bp$ and $Diff_{dis} \geq 30\ bp$, which means a local reference segment without read covering, cuteSV recognizes a deletion signature as

$$SIG_{DEL} = (Ref_{1e}, Diff_{dis}, ID) \quad (7)$$

Similarly, when $Diff_{olp} < 30\ bp$ and $Diff_{dis} \leq -30\ bp$, which means a redundant sub-read with no mapping in a local reference, cuteSV recognizes a insertion signature as

$$SIG_{INS} = \left(\frac{Ref_{1e}+Ref_{2s}}{2}, -Diff_{dis}, ID\right) \quad (8)$$

(2) If two adjacent segments on the read level are mapped to the same position on the genome level and share a local reference genome at least 30 bp in size, that is $Diff_{olp} \geq 30\ bp$, cuteSV identifies a duplication signature as

$$SIG_{DUP} = (Ref_{2s}, Ref_{1e}, ID) \quad (9)$$

(3) If two adjacent segments on the read level are aligned to the same chromosome but have divergent strands, cuteSV identifies some inversion signatures. The length of segment inverted is LEN_{INV}, which is

$$LEN_{INV} = Ref_{2e} - Ref_{1e} \quad (10)$$

Since unknown of the primary strand of an inversion, cuteSV will record all possible scenarios, such as

$$SIG_{INV} = \begin{cases} \begin{matrix}(Ref_{1e}, Ref_{2e}) \\ (Ref_{1s}, Ref_{2s})\end{matrix}, if\ LEN_{INV} \geq 50\ bp \\ \begin{matrix}(Ref_{2e}, Ref_{1e}) \\ (Ref_{2s}, Ref_{1s})\end{matrix}, if\ LEN_{INV} \leq -50\ bp \end{cases} \quad (11)$$

(4) If two adjacent segments on the read level are mapped to different chromosomes, and the distance on the read level is below 100 bp, cuteSV recognizes a translocation or BND signature based on the combination of orientation as

$$SIG_{BND} = \begin{cases} (Chr_1, Ref_{1e}, Chr_2, Ref_{2s}, ID) & , if\ "0 \leq " \\ (Chr_2, Ref_{2s}, Chr_1, Ref_{1e}, ID) & , if\ "0 \geq " \\ (Chr_1, Ref_{1e}, Chr_2, Ref_{2e}, ID) & , if\ "1 \leq " \\ (Chr_2, Ref_{2e}, Chr_1, Ref_{1e}, ID) & , if\ "1 \geq " \\ (Chr_1, Ref_{1s}, Chr_2, Ref_{2s}, ID) & , if\ "2 \leq " \\ (Chr_2, Ref_{2s}, Chr_1, Ref_{1s}, ID) & , if\ "2 \geq " \\ (Chr_1, Ref_{1s}, Chr_2, Ref_{2e}, ID) & , if\ "3 \leq " \\ (Chr_2, Ref_{2e}, Chr_1, Ref_{1s}, ID) & , if\ "3 \geq " \end{cases} \quad (12)$$

where 0, 1, 2, 3 indicates the combination of strands "++", "+-", "-+", "--", respectively, "≤" and "≥" means the chromosome ID of Chr_1 is in front of that of Chr_2 and vice versa.

(5) For some more complicated events, e.g. a read with three segments or more and containing a translocation in one of the middle segments, an insertion or duplication signature might be detected if other segments except the translocation one is aligned to the same chromosome. Specifically, referring to Eq. 6, if a larger than 30 bp overlap is shared with two segments, a duplication signal will be detected like Eq. 9. By contrast, if $Diff_{dis}$ meets less than -30 bp according to Eq. 5, an insertion signature will be generated as

$$SIG_{INS} = (\min(Ref_{1e}, Ref_{2s}), -Diff_{dis}, ID) \quad (13)$$

Clustering of SV signatures

cuteSV sorts all SV signatures by genome coordinate and category firstly. For two nearest signals $SIG_1$and SIG_2 belonging to the same SV type, cuteSV group them together when meeting

$$BK_1 - BK_2 \leq TH_{type} \quad (14)$$

where BK_1 and BK_2 are breakpoints or breakends of the two signatures, TH_{type} is a threshold that control the compactness of clustering on different SV types. In general, TH_{type} is 50 to 500 bp.

After the first-round of clustering above, cuteSV filters each candidate SV group based on quantity of signal reads. Those groups with the shortage of quantity will be discarded, which might carry the sequencing errors. For the remaining clusters, cuteSV runs a new round of clustering to refine variants and their every potential SV allele as below:

(1) For deletions and insertions, we introduce Len_bias to describe a length with specific percentage on the mean variant size of signatures, which is

$$Len_{bias} = \alpha \times \frac{1}{|Group_{SIG}|} \sum_{k \in Group_{SIG}} Size_k \quad (15)$$

where α is the percentage parameter, $Group_{SIG}$ is a given cluster, and $Size$ is the variant size of a signature in $Group_{SIG}$. cuteSV clusters in each given cluster again using Len_{bias} as a standard to divide various SV alleles.

For the subcluster with most supporting reads, cuteSV recognizes a major allele when meeting

$$\begin{cases} SupR \geq SR_{min} \\ SupR \geq \mu \times NumC \end{cases} \quad (16)$$

where $SupR$, SR_{min}, μ, and $NumC$ are the number of supporting reads, the threshold of minimum supporting reads, a specific ratio, and the total number of signatures in this group, respectively. For the rest of subclusters with more than SR_{min} supporting reads, alternative alleles are recognized by cuteSV.

However, for some cases that have no enough supporting reads for each allele, the Eq.16 above is no applicable any more. To discover these alleles, one another heuristic rule is introduced. For the largest two subclusters $SupR_{first}$ and $SupR_{second}$, cuteSV identifies the major and alternative alleles when meeting

$$\begin{cases} SR_{min} \leq SupR_{first} \leq \mu \times NumC \\ 0.4 \times NumC \leq SupR_{second} \leq SR_{min} \\ 0.95 \times NumC \leq SupR_{first} + SupR_{second} \end{cases} \tag{17}$$

which means almost all of SV signatures support the two alleles, meanwhile, both two have forty percent at least supporting signatures in this cluster.

(2) For duplications and inversions, cuteSV focuses on the diversity of start breakpoints in their first-round of clustering, that is, cuteSV will integrate the signatures if their start breakpoints are less than 500 bp apart. Since the large size and complexity of duplications and inversions, it is necessary to run a new round of clustering based on end breakpoints to figure out different alleles, even though sharing the similar start breakpoints. cuteSV recognizes corresponding alleles when meeting Eq. 16, where μ is one third for duplications and zero for inversions.

(3) For translocations, cuteSV focuses on the diversity of breakpoints in the first chromosome in the first-round of clustering and employs a more stringent TH_{type} (50 bp). Similarly, in the next round of clustering, cuteSV still utilizes 50 bp TH_{type} as well to cluster each signature on the breakpoints in another chromosome. The supporting reads in each subcluster are relatively few due to the more stringent TH_{type} and diverse combinations of chromosome and orientation. For considering the sensitivity of detection, cuteSV identifies a translocation allele when meeting Eq. 16, where μ is 0.6 and SR_{min} is half of it.

Final detection and genotyping

First of all, cuteSV predicts breakpoints and SV sizes by the weighted average of corresponding features in each allele and removes remove those SVs that less than 30 bp in size after calculation. Next, cuteSV can perform genotyping optionally. For a variant, we define $Area_{at}$ to describe an area for analysis of read counting, which is

$$Area_{at} = \left(\max(BK_1 - bias, 0), \min(BK_2 + bias, L_{REF})\right) \tag{18}$$

where BK_1 is the first breakpoint, BK_2 is the next breakpoint, $bias$ means the scope of extension (20 bp, in default), and L_{REF} indicates the maximum length of chromosome took into consideration. cuteSV counts the number of unique ID of reads that present in $Area_{at}$. The genotype of each SV mainly depends on the ratio of SV supporting reads to total reads. Referring to previous research in Sniffles, cuteSV assigns a homozygous reference allele (0/0) when the proportion less than 0.3, a homozygous variant allele (1/1) when the ratio greater than 0.8, and a specific heterozygous variant allele (0/1) for other cases.

Implementation of long-read mapping and SV calling

We utilize PBMM2 (version 1.0.0) and NGMLR (version 0.2.3) to perform long-read alignment on several long-read sequencing human sample datasets (see Supplementary Table 6). The parameter "--preset" of long-read aligners adjust along with different sequencing technologies. Samtools (version 1.9) is used for long-read extraction, sorted BAM generation, and sequencing data down-sampling.

SV calling is performed using Sniffles (version 1.0.11), PBSV (version 2.2.0), SVIM (version 0.4.3) and cuteSV (version 1.0.1). Specifically, for Sniffles, we use "-l 50 -s 2/3/3/4/5/10" for PacBio CLR datasets and "-l 30 -s 1/2/3" for PacBio HiFi reads, respectively. For PBSV, we perform SV calling with default parameters for PacBio CLR data and "--preset CCS" for PacBio HiFi data. For SVIM, the parameter "--cluster_max_distance 1.4" is employed for all datasets in this study. In terms of cuteSV, we use "-l 50 --max_cluster_bias_INS 100 --diff_ratio_merging_INS 0.2 --diff_ratio_filtering_INS 0.6 --diff_ratio_filtering_DEL0.7 -s 2/3/3/4/5/10" for PacBio CLR and ONT reads, "-l 30 --max_cluster_bias_INS 200 --diff_ratio_merging_INS 0.65 --diff_ratio_filtering_INS 0.65 --diff_ratio_filtering_DEL0.35 -s 1/2/3" for PacBio HiFi reads.

Performance evaluation

The evaluation of the HG002 human sample is run with Truvari (version 1.1), and the high confidence insertion and deletion callsets (version 0.6) are used as gold standard sets. Before evaluation, we preprocess the SV calls of each tool. For Sniffles, we discard inversions and translocations, and transform duplications to insertions. For SVIM, we delete SV calls that less than 40 in the quality score as author recommended, and transform duplications to insertions as well. For PBSV and cuteSV, we only select insertions and deletions for assessment. Then, BGZIP and TABIX are employed to compress and index the processed VCF files.

In evaluation, only SV calls belonging to 50 bp to 10 kbp in size, and contained within the GIAB high confidence regions (version 0.6) are considered for evaluating. A prediction is determined as a true-positive (TP) when meeting

$$\begin{cases} \max(comp_s - 1kbp, base_s) \leq \min(comp_e + 1kbp, base_e) \\ \min(comp_{size} + base_{size})/\max(comp_{size} + base_{size}) \geq 0.7 \\ comp_{type} = base_{type} \end{cases} \tag{19}$$

where $comp_s$, $comp_e$, $comp_{size}$ and $comp_{type}$ indicate start coordinate, stop coordinate, size and SV class of a prediction, and $base_s$, $base_e$, $base_{size}$ and $base_{type}$ are also start coordinate, stop coordinate, size and SV class of a call in truth set, respectively. On the contrary, a prediction is determined as a false-positive (FP) if unsatisfied **Eq. 19**. And a false-negative (FN) is assigned when there is a base call that cannot be detected by any SV calls.

Based on the results above, Precision or the ratio of TPs to total calls in predictions is defined as

$$Precision = \frac{TPs}{TPs+FPs} \tag{20}$$

Similarly, Recall or the ratio of TPs to total calls in truth set is defined as

$$Recall = \frac{TPs}{TPs+FNs} \tag{21}$$

F-measure (F1 score), a measurement of weighted averaging of both precision and recall, which is

$$F_measure = \frac{2 \times Precision \times Recall}{Precision+Recall} \tag{22}$$

The evaluation of the NA19240 and the NA12878 are similar to the HG002 benchmarking. We adopt **Eq.19** to assess every deletion, insertion (duplication regarded as the subset of insertion) and inversion

against the callsets generated from the study [41]. And then applied Eq.20 to 22 for summarizing the performance on SV calling.

For the evaluation of the Ashkenazi trio, in order to measure the performance of an SV caller via the pedigree between offspring and its parents, the callsets of parents (i.e. HG003 and HG004) produced by corresponding methods are adopted as truth sets. For each SV call of the son (i.e. HG002) except translocations, we use Eq. 19 to assign true-positive calls. As an arbitrary rearrangement type, a translocation or BND is considered precise when meeting

$$\begin{cases} |comp_{BK1} - base_{BK1}| \leq 1\text{kbp} \\ |comp_{BK2} - base_{BK2}| \leq 1\text{kbp} \\ comp_{chr1} = base_{chr1} \\ comp_{chr2} = base_{chr2} \end{cases} \tag{23}$$

where $BK1$, $BK2$, $chr1$, and $chr2$ are the combination of breakends and chromosomes of a call on the offspring and its parents, respectively. Therefore, the ratio of ADI can be reckoned via

$$ADI = \frac{\sum SVs\ in\ son - \sum precise\ SVs\ in\ son}{\sum SVs\ in\ son} \tag{24}$$

To evaluate the computational performance of the four SV callers under different threads, the runtime and memory using are assessed by "/usr/bin/time -v" command of Linux Operating System. In the output results of the command, "Elapsed (wall clock) time" and "Maximum resident set size" indicate the elapsed runtime and memory consumption, respectively. It is worth noting that SV calling performed by PBSV involves two steps, i.e. discover and call, we use the sum of wall clock time of both two steps as final elapsed runtime, whereas the memory consumption depends on the maximum memory using of the two runs.

For more details please see Supplementary Notes.

Authors' contributions

T.J. conceived and supervised the project. T.J. developed cuteSV. T.J., B.L. and Y.J. performed analysis. T.J. and B.L. wrote the manuscript. All authors read and approved the final manuscript.

Funding

This work has been supported by the National Key Research and Development Program of China (Nos: 2017YFC0907503, 2018YFC0910504, and 2017YFC1201201).

Availability of data and material

The human reference genome, raw sequencing data, alignments and gold standard truth callsets used in this study are available from the respective publications and websites listed in Supplementary Table 6. The implementation of cuteSV can be downloaded from GitHub (https://github.com/tjiangHIT/cuteSV).

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

References

1. Sudmant PH, Rausch T, Gardner EJ, Handsaker RE, Abyzov A, Huddleston J, Zhang Y, Ye K, Jun G, Fritz MH, et al: **An integrated map of structural variation in 2,504 human genomes.** *Nature* 2015, **526:**75-81.
2. Alkan C, Coe BP, Eichler EE: **Genome structural variation discovery and genotyping.** *Nat Rev Genet* 2011, **12:**363-376.
3. Rovelet-Lecrux A, Hannequin D, Raux G, Le Meur N, Laquerriere A, Vital A, Dumanchin C, Feuillette S, Brice A, Vercelletto M, et al: **APP locus duplication causes autosomal dominant early-onset Alzheimer disease with cerebral amyloid angiopathy.** *Nature Genetics* 2006, **38:**24-26.
4. Hedges DJ, Hamilton-Nelson KL, Sacharow SJ, Nations L, Beecham GW, Kozhekbaeva ZM, Butler BL, Cukier HN, Whitehead PL, Ma DQ, et al: **Evidence of novel fine-scale structural variation at autism spectrum disorder candidate loci.** *Molecular Autism* 2012, **3**.
5. Weischenfeldt J, Symmons O, Spitz F, Korbel JO: **Phenotypic impact of genomic structural variation: insights from and for human disease.** *Nat Rev Genet* 2013, **14:**125-138.
6. Macintyre G, Ylstra B, Brenton JD: **Sequencing Structural Variants in Cancer for Precision Therapeutics.** *Trends in Genetics* 2016, **32:**530-542.
7. Dennenmoser S, Sedlazeck FJ, Iwaszkiewicz E, Li XY, Altmuller J, Nolte AW: **Copy number increases of transposable elements and protein-coding genes in an invasive fish of hybrid origin.** *Mol Ecol* 2017, **26:**4712-4724.
8. Lupski JR: **Structural variation mutagenesis of the human genome: Impact on disease and evolution.** *Environ Mol Mutagen* 2015, **56:**419-436.
9. Chiang C, Scott AJ, Davis JR, Tsang EK, Li X, Kim Y, Hadzic T, Damani FN, Ganel L, Consortium GT, et al: **The impact of structural variation on human gene expression.** *Nat Genet* 2017, **49:**692-699.
10. Zichner T, Garfield DA, Rausch T, Stutz AM, Cannavo E, Braun M, Furlong EEM, Korbel JO: **Impact of genomic structural variation in Drosophila melanogaster based on population-scale sequencing.** *Genome Research* 2013, **23:**568-579.
11. Jeffares DC, Jolly C, Hoti M, Speed D, Shaw L, Rallis C, Balloux F, Dessimoz C, Bahler J, Sedlazeck FJ: **Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast.** *Nature Communications* 2017, **8**.
12. Levy S, Sutton G, Ng PC, Feuk L, Halpern AL, Walenz BP, Axelrod N, Huang J, Kirkness EF, Denisov G, et al: **The diploid genome sequence of an individual human.** *PLoS Biol* 2007, **5:**e254.
13. Wheeler DA, Srinivasan M, Egholm M, Shen Y, Chen L, McGuire A, He W, Chen YJ, Makhijani V, Roth GT, et al: **The complete genome of an individual by massively parallel DNA sequencing.** *Nature* 2008, **452:**872-876.
14. Yoon ST, Xuan ZY, Makarov V, Ye K, Sebat J: **Sensitive and accurate detection of copy number variants using read depth of coverage.** *Genome Research* 2009, **19:**1586-1592.
15. Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC, Zhang QY, Locke DP, et al: **BreakDancer: an algorithm for high-resolution mapping of genomic structural variation.** *Nature Methods* 2009, **6:**677-+.
16. Ye K, Schulz MH, Long Q, Apweiler R, Ning ZM: **Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads.** *Bioinformatics* 2009, **25:**2865-2871.
17. Chen K, Chen L, Fan X, Wallis J, Ding L, Weinstock G: **TIGRA: A targeted iterative graph routing assembler for**

breakpoint assembly. *Genome Research* 2014, **24:**310-317.

18. Hormozdiari F, Hajirasouliha I, Dao P, Hach F, Yorukoglu D, Alkan C, Eichler EE, Sahinalp SC: **Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery.** *Bioinformatics* 2010, **26:**i350-i357.
19. Jiang Y, Wang YD, Brudno M: **PRISM: Pair-read informed split-read mapping for base-pair level detection of insertion, deletion and structural variants.** *Bioinformatics* 2012, **28:**2576-2583.
20. Rausch T, Zichner T, Schlattl A, Stutz AM, Benes V, Korbel JO: **DELLY: structural variant discovery by integrated paired-end and split-read analysis.** *Bioinformatics* 2012, **28:**I333-I339.
21. English AC, Salerno WJ, Hampton OA, Gonzaga-Jauregui C, Ambreth S, Ritter DI, Beck CR, Davis CF, Dahdouli M, Ma S, et al: **Assessing structural variation in a personal genome-towards a human reference diploid genome.** *Bmc Genomics* 2015, **16**.
22. Tattini L, D'Aurizio R, Magi A: **Detection of Genomic Structural Variants from Next-Generation Sequencing Data.** *Front Bioeng Biotechnol* 2015, **3:**92.
23. Roberts RJ, Carneiro MO, Schatz MC: **The advantages of SMRT sequencing.** *Genome Biol* 2013, **14:**405.
24. Jain M, Olsen HE, Paten B, Akeson M: **The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community.** *Genome Biol* 2016, **17:**239.
25. Sedlazeck FJ, Lee H, Darby CA, Schatz MC: **Piercing the dark matter: bioinformatics of long-range sequencing and mapping.** *Nat Rev Genet* 2018, **19:**329-346.
26. Goodwin S, McPherson JD, McCombie WR: **Coming of age: ten years of next-generation sequencing technologies.** *Nat Rev Genet* 2016, **17:**333-351.
27. English AC, Salerno WJ, Reid JG: **PBHoney: identifying genomic variants via long-read discordance and interrupted mapping.** *BMC Bioinformatics* 2014, **15:**180.
28. Huddleston J, Chaisson MJP, Steinberg KM, Warren W, Hoekzema K, Gordon D, Graves-Lindsay TA, Munson KM, Kronenberg ZN, Vives L, et al: **Discovery and genotyping of structural variation from long-read haploid genome sequence data.** *Genome Research* 2017, **27:**677-685.
29. Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, Schatz MC: **Accurate detection of complex structural variations using single-molecule sequencing.** *Nature Methods* 2018, **15:**461-+.
30. Heller D, Vingron M: **SVIM: Structural Variant Identification using Mapped Long Reads.** *Bioinformatics* 2019.
31. Chaisson MJ, Tesler G: **Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory.** *Bmc Bioinformatics* 2012, **13**.
32. Li H: **Minimap2: pairwise alignment for nucleotide sequences.** *Bioinformatics* 2018, **34:**3094-3100.
33. Jiang T, Liu B, Li J, Wang Y: **rMETL: sensitive mobile element insertion detection with long read realignment.** *Bioinformatics* 2019, **35:**3484-3486.
34. Shao H, Ganesamoorthy D, Duarte T, Cao MD, Hoggart CJ, Coin LJM: **npInv: accurate detection and genotyping of inversions using long read sub-alignment.** *BMC Bioinformatics* 2017, **19:**261.
35. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing S: **The Sequence Alignment/Map format and SAMtools.** *Bioinformatics* 2009, **25:**2078-2079.
36. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, Weng Z, Liu Y, Mason CE, Alexander N, et al: **Extensive sequencing of seven human genomes to characterize benchmark reference materials.** *Sci Data* 2016, **3:**160025.
37. Zook JM, Hansen NF, Olson ND, Chapman LM, Mullikin JC, Xiao C, Sherry S, Koren S, Phillippy AM, Boutros PC, et al: **A robust benchmark for germline structural variant detection.** *bioRxiv* 2019**:**664623.
38. Travers KJ, Chin CS, Rank DR, Eid JS, Turner SW: **A flexible and efficient template format for circular consensus sequencing and SNP detection.** *Nucleic Acids Research* 2010, **38**.
39. Wenger AM, Peluso P, Rowell WJ, Chang PC, Hall RJ, Concepcion GT, Ebler J, Fungtammasan A, Kolesnikov A, Olson ND, et al: **Accurate circular consensus long-read sequencing improves variant detection and assembly of a**

human genome. *Nat Biotechnol* 2019.

40. Clarke L, Fairley S, Zheng-Bradley X, Streeter I, Perry E, Lowy E, Tasse AM, Flicek P: **The international Genome sample resource (IGSR): A worldwide collection of genome variation incorporating the 1000 Genomes Project data.** *Nucleic Acids Res* 2017, **45:**D854-D859.

41. Chaisson MJP, Sanders AD, Zhao X, Malhotra A, Porubsky D, Rausch T, Gardner EJ, Rodriguez OL, Guo L, Collins RL, et al: **Multi-platform discovery of haplotype-resolved structural variation in human genomes.** *Nat Commun* 2019, **10:**1784.

42. Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, Hide W, Salit M: **Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls.** *Nat Biotechnol* 2014, **32:**246-251.

43. Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, Tyson JR, Beggs AD, Dilthey AT, Fiddes IT, et al: **Nanopore sequencing and assembly of a human genome with ultra-long reads.** *Nat Biotechnol* 2018, **36:**338-345.

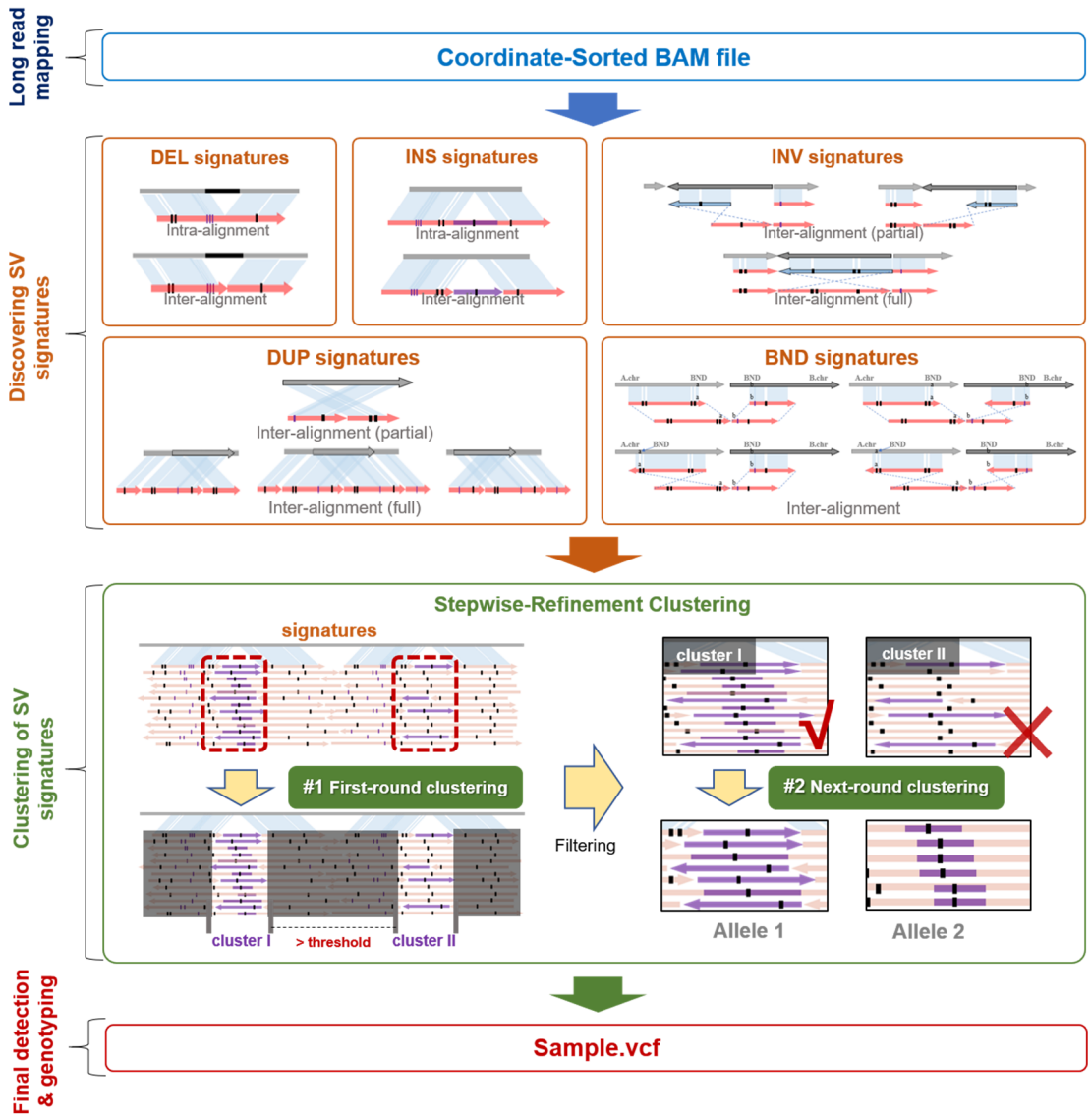

Figure 1 Scheme of cuteSV workflow.

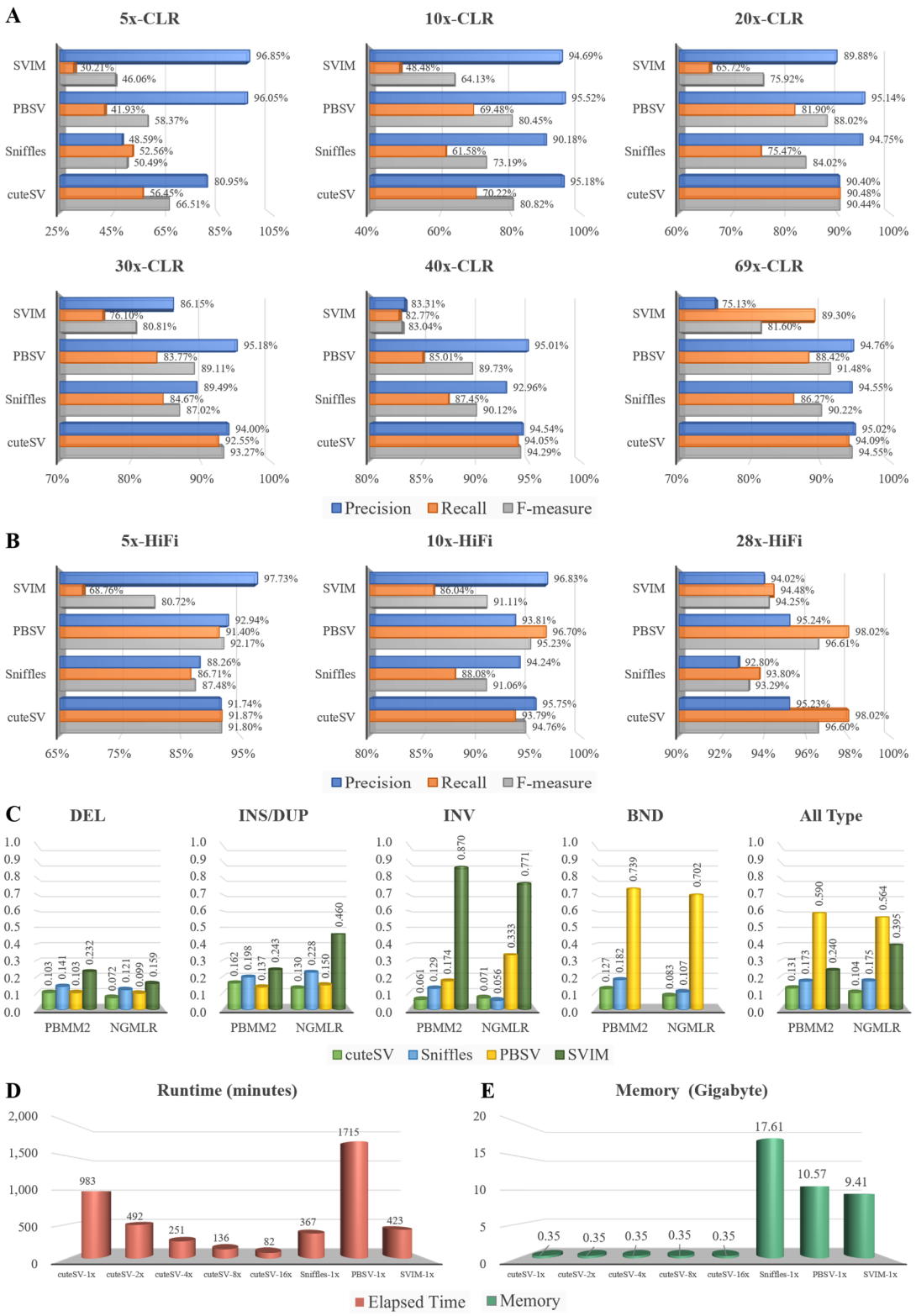

Figure 2 Performance comparison for the HG002 PacBio data with existence SV callers. (**A**) Precision, recall, and F-measure on the HG002 PacBio CLR datasets. (**B**) Precision, recall, and F-measure on the HG002 PacBio HiFi datasets. (**C**) Benchmark results of Mendelian consistency on the GIAB Ashkenazi Trio PacBio CLR datasets under different SV classes and long-read mappers. (**D**) The runtime under specific running thread. (**E**) The memory usage under specific running thread.

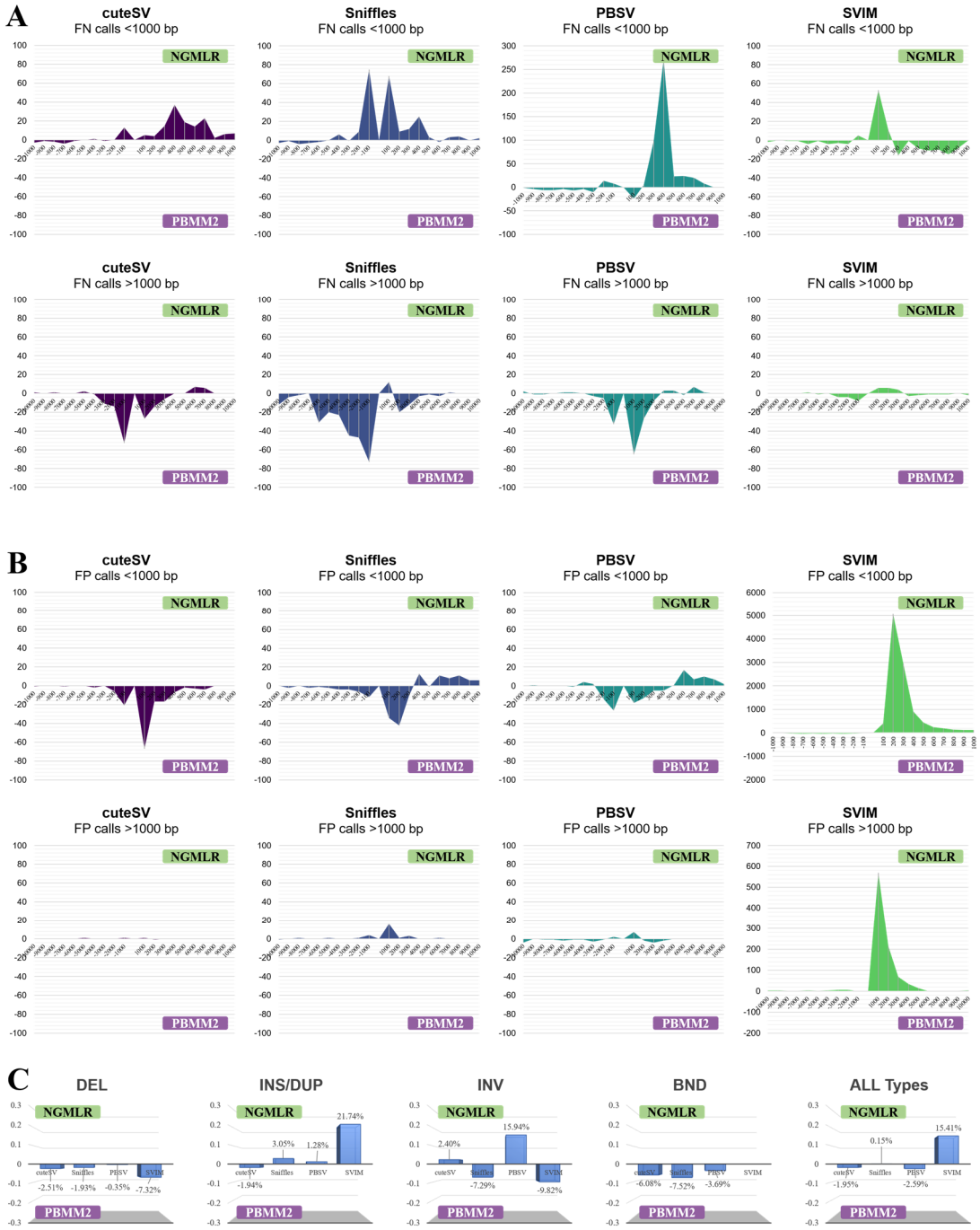

Figure 3 Performance comparison with different long-read aligners on the PacBio CLR dataset. (**A**) The amount of excess false-negative calls under different SV sizes with NGMLR and PBMM2. (**B**) The amount of excess false-positive calls under different SV sizes with NGMLR and PBMM2. (**C**) The different MDRs between NGMLR and PBMM2.

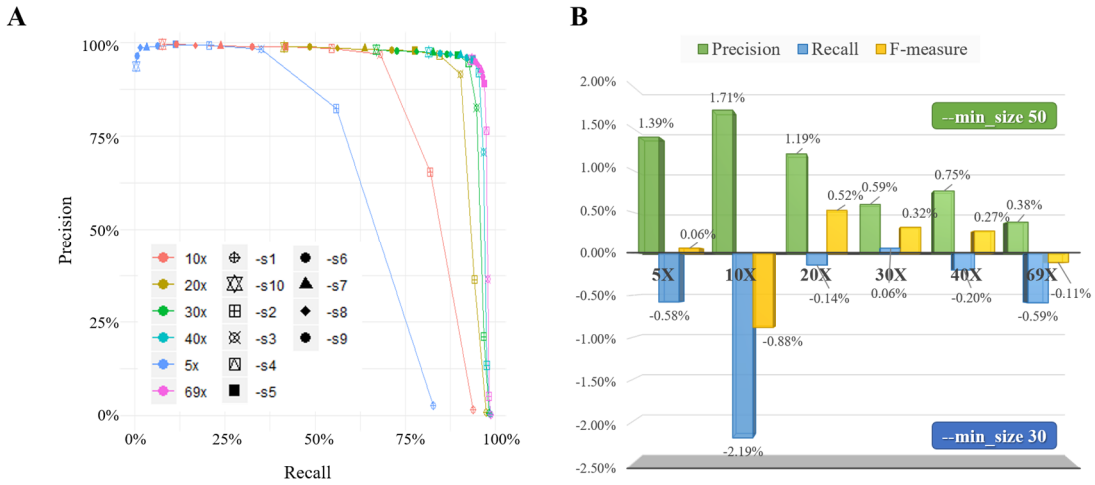

Figure 4 Comparison on the different parameters of *-s* (*--min_support*) and *–min_size* at the level of coverage. (**A**)The precision and recall under different support-read and sequencing coverages. (**B**) The difference of precision, recall, and F-measure in the best case of each coverage under *–min_size* 30 and 50.

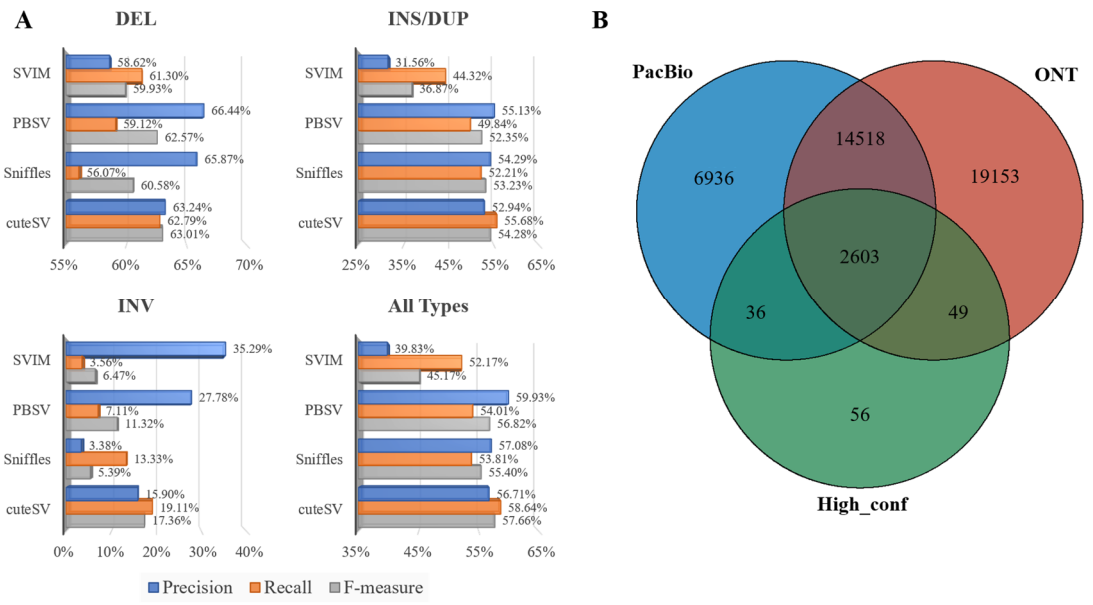

Figure 5 Performance comparison for the NA19240 and the NA12878 human samples. (**A**) Precision, recall, and F-measure on the NA19240 PacBio CLR datasets. (**B**) The Venn diagram of SV calls in cuteSV on the NA12878 PacBio CLR and ONT datasets, and the high confidence callsets.